Verification/Restart/Backup Testing For RELAP5-3D

Dr. George Mesina

RELAP5 International Users Seminar
Idaho Falls, ID
September 12-13, 2013

Outline

- Purpose
- Background – RELAP5 Testing
- Structure and Content of a Verification File
- Verification Suite, Features, and Test Matrix
- Null Testing
- Restart Testing
- Backup Testing

Purpose

- Improve Verification of RELAP5-3D
  - Establish solution metrics for verification
    - Create ability to detect small changes in solutions
  - Provide greater coverage of important code features
    - Test matrix to identify what features are covered by test cases
  - Systematize testing of processes
    - Null testing, restarts, backups
  - Automate

Background – Development of RELAP5 Testing

1. Original Testing
   a) Several key values spot-checked visually by RJW
2. Automatic installation test case report
   a) Checked that cases ran to completion and recorded number of advancements.
   b) Test set expands as new features are added
3. Diffem script – compares all printed output between two “printed output” files (the “-o” command line parameter)
4. Additional tests, such as tt=3,7,11,15 and extensive test sets added.

Background – Weaknesses

- Does not catch small changes
  - Fewer than 8 decimal places can be *checked* with diffem
- *Coverage* analysis (Understand 2.0):
  - < 75% of lines of code in relap/ directory tested by install cases
  - ~ 55% of lines of code in envrl/ directory tested by install cases

- Not all important features covered

Verification Strategy

- "Staged Verification"
  - Initial coding of a model is verified, then from version to version, model's calculation shown to be the same or changes justified.
  - Previously done via diffem on collection of test cases.
- Improve verification over what was previously done
  1. Improve coverage
     - Cover important code features
     - Select input models to test them
  2. Improve detection
     - Check most important variables
       - Governing equation primary variables
     - L_1 norm of arrays of primary variables
     - Put norms on VERIFICATION file for comparison

Detection – VERIFICATION File Values

Notation	Quantity	Symbol	Annotation	Area
Uf	Liquid internal energy	u_f	Uf	TH
Ug	Gas internal energy	u_g	Ug	TH
VOIDf	Void fraction of gas	α_g	VOIDf	TH
QUALa	Noncondensable Quality	X_a	QUALa	TH
Boron	Density of Boron	ρ_b	Boron	TH
Vf	Liquid velocity	V_f	Vf	TH
Vg	Gas velocity	V_g	Vg	TH
RHSth	RHS of Δp Linear System	b	RHSth	TH
SOLth	Pressure Drop / Velocities	Δp / (V_f, V_g)	SOLth	TH
Error	Errors	ε		Advancement
Temp	Temperature	T	Temp	Heat Transfer
Flux	Neutron Flux	φ	Flux	Neutron Kinetics
dtsum	Time steps sum	Δt, Δt_{kin}	dtsum	Advancement
Trips	Trips	T_r	Trips	Trips
Cntrl	Control System Value	Y	Cntrl	Controls
Rdc	Reductions	N/A	Rdc	Advancement
Rpt	Repeats	N/A	Rpt	Advancement

Statistical Testing Theory for Detection

- H_0: No difference between calculations of two code versions
  - $X_i = \begin{cases} 0 & \text{Case i has NO differences between VERIFICATION files of the two codes} \\ 1 & \text{otherwise} \end{cases}$
  - $X = \max \{X_i \mid i = 1, 2, \ldots, N\}$
- Test: Accept the null hypothesis if X = 0
  - Acceptance Region: {0}
  - Rejection Region for X: {1}

	H_0 is true **No differences exist**	**H_0 is false** **Differences exist**
Accept H_0	Correct Report: "No differences"	Type II Error *Don't find extant differences*
Reject H_0	**Type I Error** ***Detect non-existent differences***	Correct Report: "Differences found"

A Detection Theorem

- *Theorem - This test commits **No Type I Error**.*
  - *Type I Error is saying calculations differ when no difference exists.*
- Proof: Assume H_0 is true
  - Then there are no differences in calculations for any input deck
  - No differences in "calculations for test deck i" means $X_i = 0$
  - $P(X_i = 1 \mid \text{No difference exists for } i^{th} \text{ case}\} = 0.0$
  - Also, if no test has a difference, $X = \max \{X_i \mid i = 1, 2, \ldots, N\} = 0$
  - Thus, $\alpha = P(X = 1 \mid \text{No difference exists}\} = 0.0$
- Test will never send programmers after a phantom bug.
- *The test has a **significance level, $\alpha = 0.0$**.*
- Of course, the test must be programmed properly.

Verification File Considerations

- Does the Verification File catch every possible difference?
  - Numerical round-off? – Almost impossible with real*16 sums
  - Numerical cancellation? – None with L_1 norm
  - Bad calculation missed?
    - Secondary variables (derived from primary) errors are caught when they affect primary variables
    - Tertiary (output-only) quantity errors can be missed
- Detection is recognition of differences between two different code runs
  - Can be different versions of code on same input
  - Original run vs. restart from intermediate time
  - Original run vs. run w/ forced backup
- Other considerations
  - Identification of run
  - User control

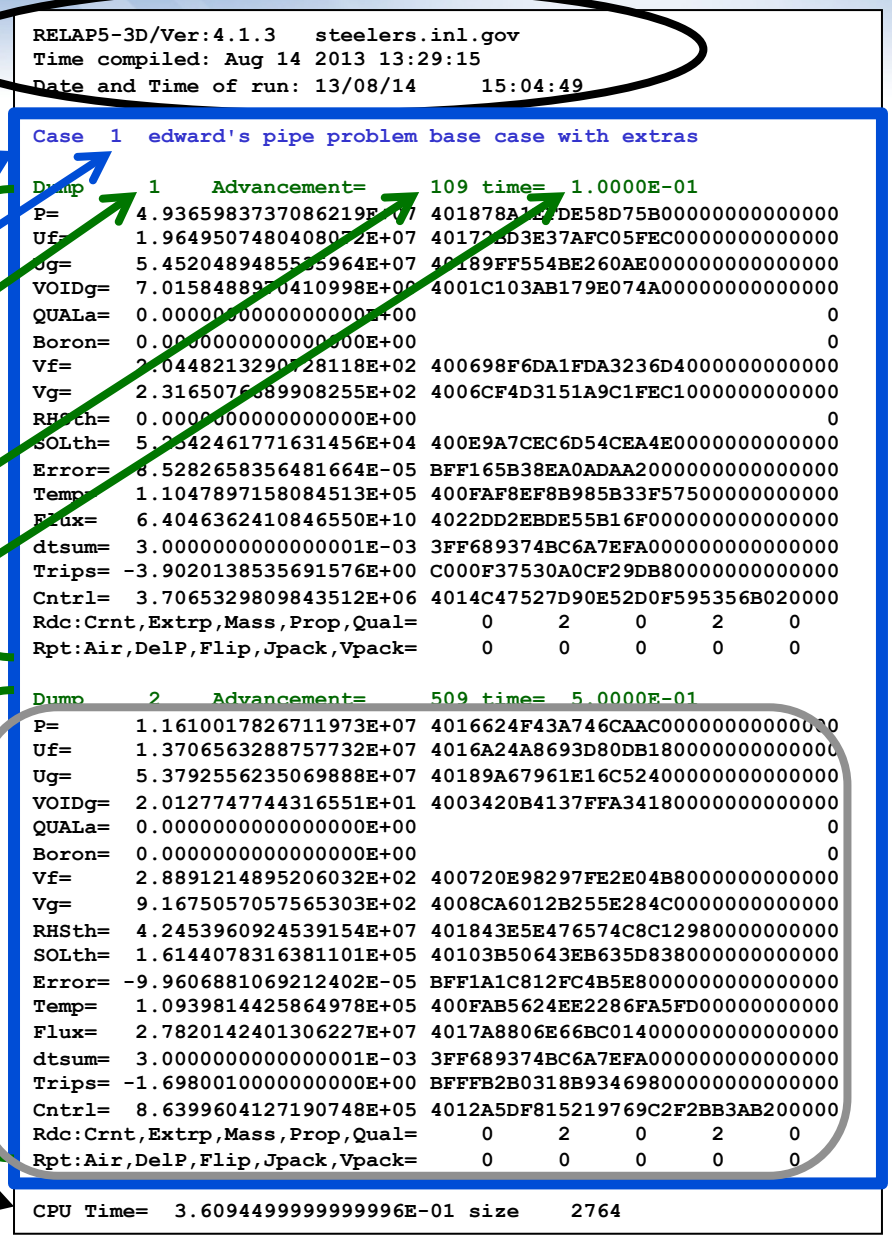
Functional Requirements for Verification File

Attribute	Description of verification file and data
1	On/off switch for verification file
2	Verification file naming by default or via input
3	When on, automatic verification dump on final time step
4	Start and end time specified via input
5	Manageable size. Less than 1 MB.
6	Unique identifiers for code version and computer name
7	Includes sums (L_1 norm) of calculated values
8	Includes execution time

- **Note:** L_1 norms are presented 2 ways:
  - Scientific Notation
  - Hexadecimal

A Verification File

- **Header**
  - Code & Computer ID
  - Data/Time Compiled
  - Date/Time Run
- **Input Case**
  - Case # & Title
- **Dump**
  - Dump #
  - Advancement #
  - Cumulative Time
  - L_1 norms
- **Footer**
  - CPU Time
  - Bytes (Upper Limit)



Verification File – Closer Look

- **Header**

```
RELAP5-3D/Ver:4.1.3    steelers.inl.gov
Time compiled: Aug 14 2013 13:29:15
Date and Time of run: 13/08/14      15:04:49
```

- **Top of Case**

```
Case  1  edward's pipe problem base case with extras
```

- **Top of Verification Dump**

```
Dump      2     Advancement=      509 time=  5.0000E-01
```

- **Footer**

```
CPU Time=  3.6094499999999996E-01 size     2764
```

Verification File Example

- Sums accumulated in Quadruple Precision (128-bits)
  - Red shows bits beyond double precision accuracy
- Left side 1pe24.16 | Right side Z32 (Hexadecimal)

P=	4.93659837370862**19**E+07	401878A1EFDE58D7**5B**0000000000000
Uf=	1.96495074804080**72**E+07	40172BD3E37AFC05**FEC**0000000000000
Ug=	5.45204894855359**64**E+07	40189FF554BE260A**E**0000000000000
VOIDg=	7.01584889704109**98**E+00	4001C103AB179E07**4A**0000000000000
QUALa=	0.00000000000000E+00	0
Boron=	0.00000000000000E+00	0
Vf=	2.04482132907281**18**E+02	400698F6DA1FDA32**36D**4000000000000
Vg=	2.31650766899082**55**E+02	4006CF4D3151A9C1**FEC**1000000000000

Verification File Theory

	H_0 is true **No differences exist**	**H_0 is false** **Differences exist**
Accept H_0	Correct Report: “No differences”	**Type II Error** ***Don’t find extant differences***
Reject H_0	Type I Error *Detect non-existent differences*	Correct Report: “Differences found”

- Goal: Reduce Type II error as much as possible.
- Two aspects
  - Detection and Coverage

Coverage

- Coverage is determined by the Test Suite
- Design Test Suite to cover as much of the code as is practical.
  - 100% coverage is impractical.
- Ignore traditional coverage measures:
  - lines of code
  - program units
- We consider code features only
- Features-Cases (Verification) matrix
  - Column 1 = Name of Feature
  - Column 2 = X if feature is tested
  - Column 3 = X if feature is restarted

Top of Feature-Cases (Verification) Matrix

Features	Present	Restart	2phspump.i	3dflow.i	ans.i	boronm.i	crit.i	cyl3.i	Drift N/A	duklerm.i	eccmix.i	edhtrkm.i	eflag.i	enclss.i	fric.i	fwhtr.i	gota27.i	hse.i	httable.i	httest.i	hxco2m.i	jetjun.i	jetpmpm.i	l2-5-emA.i	l31acc.i	neptunus20m.i	pack.i	pitch.i	radialm.i	rcpr.i	refbunm.i	reflecht.i	regime.i	rigidbodym.i	rthetam.i	rtsampnm.i	rtsamppm.i	slab3.i	sphere3.i	state.i	todcnd.i	turbine9.i	typ12002.i	typ_kindt.i	valve.i	varvol2.i
	#	#	1	2	3	4	5	6		7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
Hydrodynamic Component																																													
SNGLVOL	X	X	X							X			X			X						X	X	X						X					X	X					X	X	X		X
TMDPVOL	X	X	X	X		X	X	X		X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X
SNGLJUN	X	X	X			X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X		X	X	X	X		X	X	X	X	X			X	X	X	X	X
TMDPJUN	X	X	X			X		X		X	X			X	X	X	X		X	X	X	X	X	X		X			X			X	X	X	X	X	X	X	X				X	X		
PIPE	X	X			X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X		X	X		X	X			X	X	X			X	X	X	X	X	X	X	X	X	X	
ANNULUS	X	X																						X											X	X						X	X			
PRIZER	X	X																								X																				
BRANCH	X	X								X											X			X						X	X				X	X						X	X			
SEPARATR	X	X																						X											X	X						X	X			
Black box	X	X																																									X				
GE																																														
JETMIXER	X	X																					X																							
TURBINE	X	X																																								X				

Summary of Code Features Covered

Feature Category	Number of features
Hydro component	29
Volume flag - tlpvbfe	7
Wall friction options	6
Junction flag - jefvcahs	14
Junction form loss	4
Flow regimes	6
Heat structure type	3
Heat transfer modes	7
Heat structure BC types	8
Heat source options	5
Material Prop	3
Metal-Water	3
Subtotal	95

Feature Category	Number of features
Enclosure	2
Reactor kinetics	12
Decay Heat	11
Trips	2
Control variables	32
Tables	8
Flow regimes	6
Equation Solvers	5
Card 1 Options	9
Proprietary	5
Other Major Options	7
Subtotal	99
Total	194

Verification Matrix Generator

- A collection of Unix scripts analyzes the test cases and creates a raw matrix that is imported into MS Word for presentation
  - Input files alone insufficient
  - Input and Output file do not indicate all features tested
  - Special coding activated with verification writes info about other features
- The scripts do fill out the entire matrix.
  - Some is done manually.
- Test Suite Summary
  - 43 input decks comprised of 125 input cases
  - 194 features

3 Major Types of Testing

- Detection is recognition of differences between two different code runs
  - Remove lines with “Time” or “RELAP” before comparing
- Null Testing
  - Compare verification files from two different versions of the code (code ID differs) running the same input
- Restart Testing
  - Compare original run verification file with one created by restarting from an intermediate time
- Backup Testing
  - Compare an original run with a run with at least one forced backup
  - A backup repeats a timestep with same dt and a modified discrete system that accounts for on of the following:
    - 1) Air appearance
    - 2) Velocity flip-flop
    - 3) Water packing

199 card activates verification

- 199 card format
  - 199 Word(1) Word(2) Word(3) Word(4)
- Word(1) can be “verify” or “noverify”
  - “verify” activates verification
  - “noverify” stops verification on a “case” after verify-activation
    - No words 2, 3 and 4
- Word(2) can be:
  - dump – write verify dumps on specified steps
  - backair – backup for air (non-condensable) appearance
  - backpck – backup for water packing
  - backvel – backup for velocity flip-flop
  - backall – backup every timestep, 2 forward / 1 back

199 card

- If Word(2) is *dump*, *backair*, *backpck*, or *backvel*
  - Word(3) = start, *integer* advancement *or real* time
    - Integer start = ncount
  - Word(4) = shut off-advancement control
    - If Word(3) is an integer, Word(4) = ncount
    - If Word(3) is real, Word(4) = number of advancements
    - Word(4) = -1 means active until end of transient
- NOTE: These Word(2) choices dump for every step they are active
- If Word(2) is “backall”
  - Only verification dump is on final step.
  - For “backall”, start = successful advancement
- “199 verify backall 1 -1” performs a backup on every successful step.

Testing Directory Structure & Contents

- Testing implemented with a Directory Structure and Makefiles
- Verify directory holds:
  - Directory for each test case
  - Master Makefile
  - Auxiliary files
- Test case directory holds:
  - Null test input file
  - Restart test input file
    - E.G. for ans.i, the restart file is ans.r.i
  - Possibly an apt-plot script
- NOTE: Backup test input files are generated from Null case input to guarantee consistency
  - E.G. for ans.i, the generated backup input file is ans.bk.i

Master Makefile

- Include files for
  - Location of executable, fluids, tools, verify-target, etc.
  - Lists of input files of the test suite
    - Null, Restart, and Backup currently use same tests.
- Targets
  - Preparation, such as cleaning, linking required files to subdirectories, etc.
  - Run testing
    - Base (null test), restart, backup, or all 3
  - Report on success of: Null, restart, or backup testing

Implementation in Source Code

- New source code module verifymod.F90
  - Its internal subroutines work on its data (no use module stmts)
- Two new subroutine files: verfsum.F, verfbackup.F90
  - These main routines (and internal ones) use additional modules
- Subroutine rdebug was rewritten
- Isolated coding: almost no changes to other RELAP5 coding except to insert calls and access a few verifymod scalars
- RELAP5-3D subroutines affected: dtstep, hydro, syssol, tran, ufilfmod, ufilsmod

Results

- Null testing generated 6 User Problems
- Restart Testing generated 25 User Problems
- Backup Testing generated 37 User Problems
- Many of these relate to using a large number of input cases in restart and backup problems.
- Most have been resolved.

Additional Slides

- The following slides are not part of the presentation
- They add additional detailed information

Feature-Cases (Verification) Matrix

- Features by sections

Hydrodynamic component:			
SNGLVOL	SEPARATR	VALVE	CPRSSR
TMDPVOL	Black box	CHKVLV	MTPLJUN
SNGLJUN	GE	TRPVLV	ACCUM
TMDPJUN	JETMIXER	INRVLV	MULTID
PIPE	TURBINE	MTRVLV	SNGLFW
ANNULUS	FWHTR	SRVVLV	MTPLFW
PRIZER	ECCMIX	RLFVLV	
BRANCH		PUMP	

Feature-Cases (Verification) Matrix

- Features by sections

Volume flag tlpvbfe	Wall friction options
t - thermal stratification	Turbulent wall friction
l - mixture level	Laminar wall friction
p - water packing	Shape factor
v - vertical stratification	Viscosity ratio
b - bundle	User defined
f - wall friction	Frictionless
e – equilibibrium	

Features-Test Matrix

Junction flag jefvcahs
j (jet junction)
e (modified PV term)
f (CCFL)
Wallis
Kutataledze
Bankoff
v (HSE; 0-3)
Top offtake
Bottom offtake
Side offtake
c (choking; 0-2)
Sub-cooled
Two phase
Super-heated
a (abrupt area; 0,1,2)
h (homogeneous)
s (momentum flux, 0-3)

Junction form loss
Constant
Reynolds dependent
Abrupt area change
Connections to six faces

Flow regimes
Horizontal
Vertical pre-CHF
Vertical post-CHF
High mixing
ECC mixer
Drift flux models

Feature-Cases (Verification) Matrix

- Features by sections

Heat structure type
Rectangular
Cylindrical
Spherical
Heat transfer modes
Forced convection
Nucleate boiling
Condensation
Film boiling
Transition boiling
Reflood heat transfer
2D heat conduction

Heat structure BC types
Adiabatic
Convective
Wall temperature
Heat flux (table)
Heat flux (control variable)
HTC vs. time
HTC vs. Temp
Alternate coupling

Heat structure heat source options
Radial
Table
Control variable
Point kinetics
Nodal kinetics
Gap conductance model

Feature-Cases (Verification) Matrix

- Features by sections

Metal water reaction
Rectangular
Cylindrical
Spherical
Enclosure model
Conduction
Radiation
Alternate fluids
Noncondensable
Valve opening/closing
Boron tracking

Material properties
Built in
Input
Function
Trips
Tables
POWER
HTRNRATE
HTC-T
HTC-TEMP
TEMP
REAC-T
NORMAREA
NORMVOL

Feature-Cases (Verification) Matrix

- Features by sections

Control variables:		
SUM	STDFNCTN	TRIPUNIT
MULT	ABS	TRIPDLAY
DIV	SQRT	POWERI
DIFFRENI	EXP	POWERR
DIFFREND	LOG	PROP-INT
INTEGRAL	SIN	LAG
DELAY	COS	LEAD-LAG
FUNCTION	TAN	CONSTANT
FEEDCTL	ATAN	SHAFT
INVKIN	MIN	PUMPCTL
	MAX	STEAMCTL

Feature-Cases (Verification) Matrix

- Features by sections

Reactor kinetics	Radionuclide transport	Steady state	**Card 1 options**
Point	**Decay Heat**	**Hydro Solvers**	11 Supercrictical
SEPARABL	NO-GAMMA	BPLU	15 $\Delta t_{Courant}$
TABLE3	GAMMA	MA18 (35)	23 Godunov
TABLE4	GAMMA-AC	PGMRES (34)	41K-loss energy dissipation
TABLE3A	ANS73	**Time step options**	50 No flipflop
TABLE4A	ANS79-1	Semi-implicit	54 Void truncation
Scram (table)	ANS79-3	Nearly-implicit	55 Annular mist
Scram (control var.)	ANS94-1	Hydro-heat explicit	**Appendix K**
Power history	ANS94-4	Hydro-heat implicit	Decay heat
Nodal	ANS05-1	Δt_{kin} triplock	Metal water reaction
RAMONA	ANS05-4	Δt_{kin} extrap.	Critical flow
HWR	G factor	Movement	CHF
GEN			Post-CHF heat transfer
Control Rod			
LSOR			
Krylov			